

Dynamic conversion between remote application program graphical user interface display programs

Publication number: DE19522185

Publication date: 1996-02-01

Inventor: LONDON MITCHELL B (US); KATZ ALAN R (US);
GOODRICH DONALD W (US); ZECK STEVEN (US)

Applicant: SPACELABS MEDICAL INC (US)

Classification:

- international: **G06F3/14; G06F9/44; G06F3/14; G06F9/44;** (IPC1-7):
G06F3/153; G06F9/44; G06F13/12

- European: G06F3/14T; G06F9/44W

Application number: DE19951022185 19950619

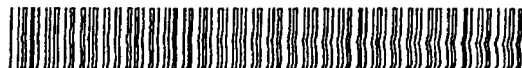
Priority number(s): US19940261492 19940617; US19950472935 19950606

Report a data error here

Abstract of **DE19522185**

The WINDOWS/NT is started up and an application called up from the procedures. It is then determined when one of the monitored items is a command for a display. If so, the WINDOWS/NT drive is suspended and an X windows manager substituted.

Data supplied from the **esp@cenet** database - Worldwide



⑨ BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ Offenlegungsschrift
⑩ DE 195 22 185 A 1

⑤ Int. Cl.⁶:
G 06 F 3/153
G 06 F 9/44
G 06 F 13/12

⑳ Aktenzeichen: 195 22 185.0
㉔ Anmeldetag: 19. 6. 95
㉕ Offenlegungstag: 1. 2. 96

DE 195 22 185 A 1

③① Unionspriorität: ③② ③③ ③①
17.06.94 US 261492 06.06.95 US 472935

⑦① Anmelder:
SpaceLabs Medical, Inc., Redmond, Wash., US

⑦④ Vertreter:
Grünecker, Kinkeldey, Stockmair & Schwanhäusser,
Anwaltssozietät, 80538 München

⑦② Erfinder:
London, Mitchell B., Redmond, Wash., US; Katz,
Alan R., Bellevue, Wash., US; Goodrich, Donald W.,
Auburn, Wash., US; Zeck, Steven, Woodinville,
Wash., US

⑤④ Ein Verfahren und System zur dynamischen Übersetzung zwischen verschiedenen grafischen Benutzeroberflächen-Systemen

⑤⑦ Die vorliegende Erfindung betrifft ein Übersetzungsverfahren, das einen computerfernen Zugriff zu einem Anwendungsprogramm, das auf einer Hauptmaschine in ihrer ursprünglichen Betriebssystem-Umgebung ausgeführt wird, schafft. Das Übersetzungsverfahren überwacht Nachrichten, die von dem Anwendungsprogramm an eine Anwendungsschnittstelle, die durch das ursprüngliche Betriebssystem geschaffen wird, gerichtet werden. Auf das Erkennen einer Nachricht hin, die eine grafische Benutzeroberfläche des ursprünglichen Betriebssystems betrifft, konvertiert das Übersetzungsverfahren die Nachricht in ein Protokoll, das von einer computerfernen, grafischen Benutzeroberfläche erkannt wird. Durch Überwachen und Konvertieren von Nachrichten auf diese Weise, ermöglicht es das Übersetzungsprogramm, daß das Anwendungsprogramm computerfern dargestellt wird.

DE 195 22 185 A 1

Technisches Gebiet

Die vorliegende Erfindung betrifft eine Laufzeitübersetzung zwischen verschiedenen grafischen Benutzeroberflächen-Systemen und umfaßt insbesondere ein Verfahren und ein System, die es erlauben, Anwendungen, die für lokale Anzeigesysteme geschrieben wurden, für computerferne Anzeigemöglichkeiten eines anderen grafischen Benutzeroberflächen-Systems zu verwenden.

Hintergrund der Erfindung

Derzeitig gibt es ungefähr 50 Millionen Computeranwender, die eine grafische Benutzeroberfläche verwenden, die als "WINDOWS" von der "MICROSOFT" Corporation bekannt ist. Derzeitig entwickelt sich sehr schnell "MICROSOFT's WINDOWS/NT" als ein bevorzugtes Betriebssystem. Als Konsequenz werden eine große Anzahl von Anwendungsprogrammen so geschrieben, daß sie mit "WINDOWS/NT" kompatibel sind. "WINDOWS/NT" stellt eine einzelne grafische Benutzeroberfläche zur Verfügung, an die sich all diese Anwendungsprogramme halten müssen. Außerdem stellt ein Client/Server-Windowssystem, das als "X" bekannt ist, und anfänglich am Massachusetts's Institute of Technology entwickelt wurde, ein Protokoll (X-Protokoll) zur Verfügung, das es erlaubt, Anwendungsprogramme computerfern auf einer oder mehreren grafischen Benutzeroberflächen (beispielsweise Motif, Open-Look) darzustellen. Unglücklicherweise sind die Anwendungsprogramme, die für das "WINDOWS/NT"-Betriebssystem entwickelt worden sind, nicht mit dem X-Protokoll kompatibel. Als Konsequenz haben Verwender des X-Client/Server-Windowssystems keinen bequemen Zugriff auf diese Anwendungsprogramme. Dies ist unglücklich, weil viele der Anwendungsprogramme, die für das "WINDOWS/NT"-Betriebssystem entwickelt worden sind, was die Kosten und die Merkmale betrifft, den für das X-Client/Server-Windowssystem entwickelten überlegen sind.

Zusammenfassung der Erfindung

Die vorliegende Erfindung betrifft Übersetzungsverfahren, die einen computerfernen Zugriff zu einem Anwendungsprogramm schaffen. Insbesondere stellt das Übersetzungsverfahren einen computerfernen Zugriff auf ein Anwendungsprogramm, das auf einem Hauptrechner in seiner ursprünglichen Betriebssystemumgebung ausgeführt wird, zur Verfügung. Das Übersetzungsverfahren überwacht Nachrichten, die von dem Anwendungsprogramm an eine Anwendungsschnittstelle gerichtet werden, die durch das ursprüngliche Betriebssystem zur Verfügung gestellt wird. Auf das Erkennen einer Nachricht, die eine grafische Benutzeroberfläche des ursprünglichen Betriebssystems betrifft, hin, konvertiert das Übersetzungsverfahren die Nachricht in ein Protokoll, das durch eine computerferne grafische Benutzeroberfläche erkannt wird. Das Übersetzungsverfahren überwacht und konvertiert auch Befehle auf der Treiberebene. Durch Überwachen und Konvertieren von Nachrichten auf diese Weise, erlaubt das Übersetzungsverfahren, daß das Anwendungsprogramm computerfern dargestellt wird.

Diese und weitere Vorteile der Erfindung ergeben sich aus der taillierten Beschreibung bevorzugter Ausführungsformen der Erfindung unter Bezugnahme auf die Zeichnungen. Es zeigen:

5 Fig. 1 eine bevorzugte Ausführungsform der vorliegenden Erfindung,

Fig. 2 einen Vergleich der Systemdesignmerkmale für ein Client/Server-Windowssystem und ein Betriebssystem,

10 Fig. 3 ein Flußdiagramm der Verfahrensschritte der bevorzugten Ausführungsform der vorliegenden Erfindung,

Fig. 4 ein Flußdiagramm der Verfahrensschritte eines Anwendungsprogramms (Stand der Technik),

15 Fig. 5 ein Flußdiagramm der Verfahrensschritte für die Wechselwirkung zwischen dem Anwendungsprogramm und einem ursprünglichen Betriebssystem (Stand der Technik),

20 Fig. 6 ein Flußdiagramm der Verfahrensschritte für ein Beispiel gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung.

Detaillierte Beschreibung der Erfindung

25 Die vorliegende Erfindung betrifft ein Übersetzungsverfahren, daß es erlaubt, ein für ein bestimmtes grafisches Benutzeroberflächensystem (GUI) geschriebenes Programm auf einem anderen grafischen Benutzeroberflächensystem mit der Laufzeit darzustellen. Das GUI, auf dem das Anwendungsprogramm dargestellt wird, ist als Ziel-GUI bezeichnet. Falls das Ziel-GUI in der Lage ist, über ein Netzwerk computerfern darzustellen, erlaubt es das Übersetzungsverfahren, daß das Anwendungsprogramm computerfern dargestellt wird, und mit ihm computerfern interagiert wird, während es innerhalb seines ursprünglichen (nicht-computerfernen) GUI ausgeführt wird.

30 In einer bevorzugten Ausführungsform sorgt die vorliegende Erfindung für eine Übersetzung von Windows und "WINDOWS/NT" in das X-Windowssystem und ermöglicht somit eine computerferne Darstellung von Anwendungen, die für "MICROSOFT WINDOWS" und "WINDOWS/NT" entwickelt wurden, auf X-Terminals und X-Servern. Insbesondere erlaubt die bevorzugte Ausführungsform, daß jede derartige Anwendung als ein vollständig kompatibles X-Client-Programm zu computerfernen X-Servern erscheint. Der Term "X-Client" betrifft ein Anwendungsprogramm, daß das vorher beschriebene X-Protokoll verwendet. Der Ausdruck "X-Terminal" betrifft irgendeine Anzeigekomponente (beispielsweise ein alleinstehendes Terminal oder ein Arbeitsplatz), der eine Anzeigekomponente eines X-Client/Server-Windowssystems ist. Der Ausdruck "X-Server" betrifft das Verfahren, das die Anzeigehardware (beispielsweise eine Bildschirmanzeige, eine Tastatur oder eine Maus) steuert und ist verantwortlich für die Durchführung der Zeichnung und der Bildaufbereitung der Bildschirmanzeige. Ein attraktives Merkmal eines X-Client-Programms ist es, daß es auf irgendeinem X-Terminal dargestellt und/oder mit ihm interagiert werden kann (beispielsweise über eine Maus, eine Tastatur, etc.), vorausgesetzt, daß eine geeignete Kommunikationsverbindung (typischerweise, allerdings keine Einschränkung darstellend, das unten beschriebene TCP/IP-Protokoll) vorhanden ist, das die Maschine, auf der der X-Client läuft, und das X-Terminal verbindet.

65 In dem X-Client/Server-Windowssystem kommuniziert ein X-Client mit der zugrundeliegenden Hardware

(beispielsweise Bildschirm, Maus, Tastatur, etc.) über den X-Server. Ein getrennter X-Client, als Windowmanager bekannt, koordiniert mit dem X-Server die Verwaltung der Darstellung. Insbesondere verfolgt der Windowmanager die Zustände aller X-Windows. Der Ausdruck "X-Window" betrifft irgendeinen Abschnitt einer Anzeige eines X-Terminals, das seine eigenen Daten (beispielsweise ein Dokument oder eine Nachricht) enthält. Ein wesentlicher Vorteil des X-Client/Server-Windowmanagers ist es, daß es einen X-Clienten nicht auf die Verwendung eines bestimmten Windowmanagers beschränkt. Statt dessen kann ein Benutzer den Windowmanager auswählen, von dem er glaubt, daß er das attraktivste Aussehen und Empfinden bietet.

Fig. 1 stellt eine bevorzugte Ausführungsform der vorliegenden Erfindung dar. Ein Hauptrechner 110 führt Anwendungsprogramme, beispielsweise "MICROSOFT EXCEL" 120 und "WORD FOR WINDOWS" 130 unter dem "WINDOWS/NT"-Betriebssystem 140 aus. Das "WINDOWS/NT"-Betriebssystem sowie die Anwendungsprogramme 120, 130 sind in einem Hauptspeicher des Hauptrechners enthalten. Insbesondere führt eine in dem Hauptrechner 110 enthaltene CPU die Anwendungen sowohl des "WINDOWS/NT"-Betriebssystems 140 als auch der Anwendungsprogramme 120, 130 aus. Das "WINDOWS/NT"-Betriebssystem schafft eine Anwendungsprogrammchnittstelle (API), die diesem Anwendungsprogramm 120, 130 erlaubt, mit dem Betriebssystem 140 zu kommunizieren. Zusätzlich liefert das "WINDOWS/NT" Betriebssystem auch eine grafische Standardbenutzeroberfläche (GUI), die das Aussehen und die Empfindung für alle unter "WINDOWS/NT" laufenden Anwendungen definiert. Diese grafische Standardbenutzeroberfläche ist so ausgelegt, daß sie Befehle von dem Anwendungsprogramm empfängt. Kurz gesagt, verwenden die Anwendungsprogramme 120, 130 die Anwendungsschnittstelle, um vom Betriebssystem die Durchführung einer Vielfalt von Aufgaben anzufordern. Beispielsweise umfassen repräsentative Aufgaben das Öffnen und Schließen von Dateien sowie die Durchführung verschiedener Einstellungen der grafischen Benutzeroberfläche. Allerdings ermöglicht das "WINDOWS/NT"-Betriebssystem keine computerferne Wechselwirkung mit Anwendungen. In seiner bevorzugten Ausführungsform überwindet die vorliegende Erfindung diese Beschränkung von "WINDOWS/NT", indem es erlaubt, Anwendungsprogramme 120, 130 auf dem Hauptrechner auszuführen, während sie computerfern auf einem X-Terminal 150 dargestellt werden. Auf diese Weise erlaubt die vorliegende Erfindung den Benutzern des X-Client/Server-Windowmanagers, einen Zugriff auf Anwendungen, die für das "WINDOWS/NT"-Betriebssystem entwickelt wurden. Außerdem stellt, da die bevorzugte Ausführungsform der vorliegenden Erfindung es ermöglicht, die Anwendungsprogramme 120, 130 in ihren ursprünglichen Betriebssystemumgebungen (beispielsweise dem "WINDOWS/NT"-Betriebssystem 140) auszuführen, die vorliegende Erfindung einen derartigen Zugriff in einer sehr effizienten Weise zur Verfügung.

Wie dargestellt, ist der Hauptcomputer 110 mit den X-Terminals 150 über eine Netzwerkverbindung, wie beispielsweise dem in Ethernet verwendeten TCP/IP-Protokoll, verbunden. Kurz gesagt, besteht dieses Protokoll aus einer Reihe von Schichten. Insbesondere sind die fünf untersten Schichten in aufsteigender Reihenfolge das Kabel, das Ethernet, das Internet-Protokoll (IP-Schicht), das Transportsteuerprotokoll (TCP-Schicht)

und das X-Protokoll. Diese Netzwerkverbindung ist konzeptionell durch das Bezugszeichen 160 dargestellt.

Wie oben beschrieben verwendet das X-Client/Server-Windowssystem zwei Komponenten: einen X-Server 170 und einen Windowmanager 180. Der Windowmanager 180 verwaltet die X-Windows, indem er verfolgt, wieviele X-Windows derzeit dargestellt werden, wo die X-Windows derzeit dargestellt werden etc. Da die vorliegende Erfindung die "WINDOWS/NT"-Anwendung in einen vollständig kompatiblen X-Client konvertiert, kann ein Benutzer einen bestimmten Windowmanager 180 (beispielsweise Motif, Open-look, Tom's Windowmanager) zur Verwendung mit der vorliegenden Erfindung wählen. Kurz gesagt empfängt der Windowmanager 180 Befehle, die die Darstellung auf den X-Terminals 150 betreffen. Da der Windowmanager 180 in dem X-Terminal selbst oder auf dem Netzwerk enthalten sein kann, kann der Windowmanager Befehle über die Netzwerkverbindung 160 oder direkt von dem X-Terminal 150 empfangen.

Fig. 2 vergleicht den Systemdesignaufbau für ein Client/Server-Windowssystem und ein Betriebssystem. Insbesondere vergleicht Fig. 2 den Systemdesignaufbau des X-Client/Server-Windowmanagers und des "WINDOWS/NT"-Betriebssystems. Eine Kenntnis dieser Strukturen ist bei der Diskussion der in den Fig. 3-6 beschriebenen Verfahrensschritte hilfreich.

In "WINDOWS/NT" sind Programme (beispielsweise Anwendungsprogramme) über Gerätetreiber 210 mit einer Standardschnittstelle zur Hardware 205 (beispielsweise einer Anzeigeeinrichtung, einer Tastatur, einer Maus etc.) versehen. Die verschiedenen Komponenten 215 des "WINDOWS/NT"-Betriebssystems sind über den Gerätetreiber gelegt. Eine derartige Komponente ist die Grafiktreiberschnittstelle 220 (GDI). Die GDI erlaubt, daß Anwendungsprogramme grafische Informationen ohne Bezugnahme auf die zugrundeliegende Hardware beschreiben. Die Anwendungsprogramm-schnittstelle 225 (API) existiert über der "WINDOWS/NT"-Komponentenschicht. Die API umfaßt die Konzepte von Fenstern (Windows), Ikonen, Positionsanzeiger und andere Mittel. "WINDOWS/NT" ist ein anzeigegesteuertes, auf einer Nachricht basierendes System. Verschiedene Programme kommunizieren miteinander und mit dem "WINDOWS/NT" durch Weitergabe von Nachrichten über die API. In bezug auf grafisch dargestellte Information verwendet die API Konstruktionen auf hoher Ebene, wie beispielsweise Fenster (Windows) oder Ikonen. Im Gegensatz hierzu verwendet der Bildschirmanzeigegerätetreiber Darstellungen grafischer Daten (beispielsweise Bitmap-Daten) auf niedriger Ebene. Ein Zweck der Bildschirmanzeigegerätetreiber ist es, diese Bitmap-Daten zur Anzeigeeinrichtung zu übertragen. Beispielsweise werden, wenn der Benutzer ein Fenster bewegt oder seine Größe verändert, eine große Anzahl von BitBlts (Blockübertragungen von Bitmap-Daten) durch den Anzeigeeinrichtungstreiber ausgeführt.

Das X-Server/Client-Windowssystem ist etwa ähnlich wie das "WINDOWS/NT"-Betriebssystem strukturiert. Ein X-Server 230 stellt eine Standardschnittstelle zur zugrundeliegenden Hardware 235 zur Verfügung. Eine Bibliothek, Xlib 240, stellt eine Programmierschnittstelle (auf einer niedrigen Ebene) für X zur Verfügung. Wenn der X-Client lokal läuft, gibt der X-Client Befehle aus der Xlib-Bibliothek 240 aus, um durch den X-Server 230 auf die lokale Hardware 235 zuzugreifen. In ähnlicher Weise gibt der X-Client, wenn er computerfern

läuft, Kommandos aus der Xlib-Bibliothek 240 aus, um auf die computerferne Hardware zuzugreifen. In diesem Fall werden Netzwerkdatenpakete, die Elemente des X-Protokolls darstellen, zum X-Server 230, der die computerferne Hardware 235 bedient, geschickt. Wie "WINDOWS/NT" ist X ein anreizgesteuertes System und ein X-Client kann von dem X-Server 230 anfordern, daß er X-Ereignisse von Interesse (wie beispielsweise, wenn die Maus in ein Fenster eintritt oder es verläßt, der Benutzer etwas in einem Fenster anklickt, usw.) abschickt. Wie oben erwähnt, ermöglicht ein spezielles X-Client-Programm, der Windowmanager 245, dem Benutzer, die dargestellten Windows zu verwalten, Menüs zu definieren usw. Neben der Xlib-Schnittstelle auf relativ niedriger Ebene liefert ein X-Toolkit 250 (eine X-Werkzeugausrüstung) eine Programmierschnittstelle auf höherer Ebene. Es gibt viele derartige Toolkits, die kommerziell erhältlich sind (beispielsweise die "XT"-, "ANDREW"- und "MOTIF"-Toolkits). Jedes Toolkit wird mit einer Sammlung von Produkten Widgets 255, die eine nützliche Sammlung von X-Objekten (beispielsweise einen rollbaren Text Window Widget) sind. Nachdem ein Vergleich der Strukturen des X-Client-Windowssystems und des "WINDOWS/NT"-Betriebssystems gegeben worden sind, werden nun die Verfahrensschritte einer bevorzugten Ausführungsform der vorliegenden Erfindung beschrieben.

Fig. 3 stellt die Prozeßschritte einer bevorzugten Ausführungsform der vorliegenden Erfindung dar. Insbesondere gibt Fig. 3 einen Überblick über die Verfahrensschritte, die von dem Übersetzungsverfahren durchgeführt werden, um ein in seiner ursprünglichen Umgebung ablaufendes Anwendungsprogramm in einen vollständig kompatiblen Client eines nicht-ursprünglichen Windowssystems zu übersetzen. Zuerst wird ein Anwendungsprogramm in seiner ursprünglichen Umgebung (Schritt 305) ausgeführt. Beispielsweise ist in einer bevorzugten Ausführungsform diese ursprüngliche Umgebung das "WINDOWS/NT"-Betriebssystem. Wenn das Anwendungsprogramm in seiner ursprünglichen Umgebung ausgeführt wird, überwacht das Übersetzungsverfahren die Systemaufrufe, die die Anwendung an die ursprüngliche Umgebung über die Anwendungsprogrammchnittstelle (Schritt 310) der ursprünglichen Umgebung richtet. Beispielsweise überwacht in einer bevorzugten Ausführungsform das Übersetzungsverfahren die Befehle, die von dem Anwendungsprogramm an die durch "WINDOWS/NT" zur Verfügung gestellte Anwendungsprogrammchnittstelle gerichtet werden.

Für jeden überwachten Systemaufruf bewertet das Übersetzungsverfahren, ob der durch den API-Aufruf dargestellte Befehl das Windowmanagement betrifft. Ein Befehl "betrifft das Windowmanagement", wenn die Ergebnisse des Befehls die Positionierung des Anwendungsprogramm-Fensters verändert (beispielsweise bewegt, die Größe verändert, das Anwendungsprogramm-Fenster in einen Stapelspeicher ablegt). Somit wirkt Schritt 315 als eine Abfangeinrichtung auf hoher Ebene zwischen dem Anwenderprogramm und dem ursprünglichen Betriebssystem.

Wenn der API-Befehl das Windowmanagement betrifft, suspendiert das Übersetzungsverfahren vorzugsweise die Fähigkeit des ursprünglichen Gerätetreibers (beispielsweise des Maustreibers, des Tastaturtreibers, des Bildschirmtreibers), den Befehl zu verarbeiten (Schritt 315, JA-Pfad und Schritt 320) aus. In einigen Fällen ist das Suspendieren der Fähigkeit des ursprüng-

lichen Gerätetreibers, den Befehl zu verarbeiten, vorteilhaft, da es die Response-Zeit des Systems stark vergrößert. Durch das Anhalten des ursprünglichen Bildschirmtreibers während der Bewegung eines Fensters, zeichnet der ursprüngliche Gerätetreiber beispielsweise das Bild des Fensters nicht kontinuierlich nach, wenn der Benutzer das Fenster über den Bildschirm zieht. Als Ergebnis schickt der Bildschirmtreiber nicht kontinuierlich die BitBlts (Blöcke von Bitmap-Daten), die das Bild des Fensters darstellen, an die Bildschirmhardware. Konsequenterweise und insbesondere in Systemen, in denen eine lokale Darstellung der Ausführung der Anwendung in ihrer ursprünglichen Umgebung nicht erwünscht ist, wird die Systemperformance aufgrund der Reduktion einer unnötigen Datenübertragung stark verbessert. Nach Anhalten des ursprünglichen Gerätetreibers konvertiert das Übersetzungsverfahren den Befehl in ein Format, das durch das Ziel-GUI (Schritt 325) erkennbar ist. In einer bevorzugten Ausführungsform hält sich dieses Ziel-GUI an ein Protokoll, das als das X-Protokoll bekannt ist. Als Konsequenz konvertiert das Übersetzungsverfahren den API-Befehl in einen Befehl des X-Protokolls (beispielsweise einen Befehl aus der Xlib-Bibliothek). Wenn der Befehl konvertiert worden ist, gibt das Übersetzungsverfahren den Befehl an das Ziel-GUI-System weiter (Schritt 330). In einer bevorzugten Ausführungsform wird dieses Kommando beispielsweise an einen Windowmanager weitergegeben. Der Windowmanager führt dann diesen Befehl über den X-Server aus. Als optionalen Schritt kann das Übersetzungsverfahren den Befehl auch an das ursprüngliche GUI-System weitergeben, so daß das Windowmanagement auf einer lokalen ursprünglichen Anzeigeeinrichtung ausgeführt wird (Schritt 335). Ob der optionale Schritt durchgeführt wird oder nicht, gibt das Übersetzungsverfahren den ursprünglichen Gerätetreiber aus seinem angehaltenen Zustand frei (Schritt 335).

Wieder Bezug nehmend auf Schritt 315 gibt das Übersetzungsverfahren, wenn der überwachte Befehl das Windowmanagement nicht betrifft, den Befehl an das ursprüngliche Betriebssystem des Anwendungsprogramms weiter (Schritt 315, NEIN-PFAD, und Schritt 340). Nachfolgend fängt das Übersetzungsverfahren jeden Systemaufruf auf der Gerätetreiberbene ab und bestimmt, wann der Systemaufruf den internen Zustand des Anwendungsprogramm-Fensters beeinflußt (Schritt 340 und 345). Somit arbeiten die Schritte 340 und 345 als eine Abfangeinrichtung auf niedriger Ebene zwischen dem Anwendungsprogramm und dem ursprünglichen Betriebssystem. Für den Fall, in dem der Systemaufruf den internen Zustand des Anwendungsprogramm-Fensters nicht beeinflußt, ermöglicht das Übersetzungsverfahren, daß das Anwendungsprogramm auf herkömmliche Weise ausgeführt wird, (Schritt 345, NEIN-PFAD, und Schritt 305). Wenn der Systemaufruf den internen Zustand des Anwendungsprogramms-Fensters beeinflußt, konvertiert das Übersetzungsverfahren andererseits den Systemaufruf in einen Ziel-GUI-Aufruf nachdem der ursprüngliche Gerätetreiber angehalten worden ist (Schritt 345, JA-PFAD, Schritte 320 und 325). Nach dem Übersetzen des Systemaufrufs gibt das Übersetzungsverfahren den übersetzten Aufruf an das Ziel-GUI-System weiter (Schritt 330). Das Übersetzungsverfahren gibt dann optional den Befehl an den ursprünglichen GUI-Gerätetreiber weiter (Schritt 335). Wiederum wird dieser optionale Schritt nur durchgeführt, wenn eine lokale Anzeige der Ausführung des Anwendungsprogramms in seiner ursprünglichen Umgebung er-

wünscht ist. Durch Wiederholen der Verfahrensschritte 305 bis 345 ermöglicht die vorliegende Erfindung, daß das Anwendungsprogramm in seiner ursprünglichen Umgebung ausgeführt wird, während es gleichzeitig erlaubt, daß mit der Anwendung computerfern auf einem computerfernen Terminal oder einem Arbeitsplatz interagiert wird. Nachdem die Verfahrensschritte einer bevorzugten Ausführungsform der vorliegenden Erfindung beschrieben worden sind, wird nun die bevorzugte Ausführungsform mittels eines Beispiels erklärt.

Fig. 4 stellt die Verfahrensschritte eines Anwendungsprogramms dar. Dieses Programm ist entwickelt, um in Verbindung mit einem Betriebssystem ein Fenster zu erzeugen, das den Text "Hallo" darstellt. Die Durchführung des Programms beginnt durch Registrieren einer Windowklasse (Fensterklasse) mit ihrem erwünschten ursprünglichen Betriebssystem (Schritt 405). Diese Windowklasse beschreibt eine Reihe von globalen Attributen für eine Klasse von Fenstern (Windows) (beispielsweise eine Ikone für das Fenster, die erscheint, wenn das Fenster minimiert wird, die Hintergrundfarbe für das Fenster etc.). Wenn eine Klasse von Fenstern (Windows) in dem Betriebssystem eingetragen worden ist, erzeugt das Programm ein Fenster für die Klasse unter Verwendung eines Create Window-API-Aufrufs (Schritt 410). Der Create-Window-API-Aufruf erzeugt das erwünschte Fenster intern in dem Speicher des Computersystems. Um das Window auf der Bildschirmanzeigeeinrichtung darzustellen, gibt das Programm einen ShowWindow-API-Aufruf ab (Schritt 420). Eine Verfahrensschleife, kollektiv durch die Blöcke 430, 435, 440 und 450 dargestellt, fordert dann das Zeichnen des Texts "Hallo" auf der Bildschirmanzeigeeinrichtung an. Die Nachrichtenschleife 430 verweist auf den Code, der Nachrichten aus einer Nachrichtenwarteschlange, die durch das ursprüngliche Betriebssystem zur Verfügung gestellt wird, extrahiert. Insbesondere fordert, wenn das Programm eine Zeichennachricht (Schritt 435, JA-PFAD) empfängt, das Programm die Bildaufbereitung der Zeichenfolge "Hallo" in der Mitte des dargestellten Windows an (Schritt 440). Wenn die empfangene Nachricht keine Zeichennachricht ist (Schritt 435, NEIN-PFAD), bestimmt das Programm andererseits, ob die empfangene Nachricht eine Nachricht zum Beenden ist oder nicht (Schritt 450). Wenn die empfangene Nachricht keine Nachricht zum Beenden ist (Schritt 450, NEIN-PFAD), dann führt das Programm eine Schleife zurück zu Schritt 430 aus und wartet auf die nächste Nachricht. Wenn das Programm eine Nachricht zum Beenden empfängt (Schritt 450, JA-PFAD), endet das Programm.

Fig. 5 stellt ein Flußdiagramm der Verfahrensschritte für die Wechselwirkung zwischen dem Anwendungsprogramm von Fig. 4 und einem ursprünglichen Betriebssystem dar. Das ursprüngliche Betriebssystem antwortet auf die Registrierung einer neuen Windowklasse durch Speichern der Klassendaten für die neue Datenklasse (Schritt 510). Durch Abspeichern dieser Daten kann ein Window oder eine Anzahl von Windows basierend auf den gespeicherten Klassendaten erzeugt werden. Als nächstes antwortet das ursprüngliche Betriebssystem auf den Create-Window-API-Aufruf durch Erzeugung einer Windowstruktur (Schritt 520). Insbesondere erzeugt das ursprüngliche Betriebssystem basierend auf den von dem Create-Window-API-Aufruf übergebenen Parametern (beispielsweise dem Windowstil, der Windowüberschrift, der anfänglichen Positionierung etc.) eine Windowstruktur und gibt dem An-

wendungsprogramm eine Handhabe für die erzeugte Windowstruktur zurück. Nachdem die Windowstruktur erzeugt ist, stellt das ursprüngliche Betriebssystem als Antwort auf den ShowWindow-API-Aufruf (Schritt 530) das Window auf der Bildschirmanzeigeeinrichtung des Computersystems dar. Kurz gesagt, schafft das ursprüngliche Betriebssystem dies durch Empfangen der Handhabe als Eingabeparameter von dem Anwendungsprogramm, wobei die Handhabe das Window kennzeichnet, das das Anwendungsprogramm darstellen will. Das ursprüngliche Betriebssystem empfängt ebenfalls einen Eingabebefehl, der anzeigt, ob das Window in normalem oder minimiertem Format dargestellt werden soll. Nachdem das ursprüngliche Betriebssystem das Window dargestellt hat, beginnt das Anwendungsprogramm, Nachrichten von der Nachrichtenwarteschlange über die vorher beschriebene Verfahrensschleife (Schritte 410–450 in Fig. 4) einzugeben. Das ursprüngliche Betriebssystem stellt eine Default-Windowprozedur (Schritt 540 in Fig. 5) zur Verfügung, die alle Nachrichten, die das Anwendungsprogramm selbst nicht verarbeitet (d. h., alle Nachrichten mit Ausnahme der Nachrichten zum Zeichnen und Beenden) bearbeitet. Wie vorher beschrieben, gibt das Anwendungsprogramm, wenn es eine Zeichennachricht empfängt, einen Befehl zum Zeichnen von "Hallo" an das ursprüngliche Betriebssystem. Das ursprüngliche Betriebssystem antwortet auf diesen Befehl durch Aufruf des Bildschirmanzeigeeinrichtungstreiber (Schritt 550). Der Bildschirmanzeigeeinrichtungstreiber seinerseits gibt den Text an die Bildschirmanzeigeeinrichtung ab (Schritt 560). Auf die oben beschriebene Weise wechselwirkt das ursprüngliche Betriebssystem mit dem Anwendungsprogramm, so daß ein lokales Window dargestellt wird.

Fig. 6 stellt ein Flußdiagramm der Verfahrensschritte für ein Beispiel gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung dar. Dieses Beispiel illustriert, wie eine bevorzugte Ausführungsform für eine computerferne Interaktion in einer nicht-ursprünglichen Umgebung mit einer Ausführung einer Anwendung in der ursprünglichen Umgebung sorgt. Wie in dem Flußdiagramm gesehen werden kann, erlaubt die bevorzugte Ausführungsform, daß Systemaufrufe, die nicht mit dem Windowmanagement in Beziehung stehen (beispielsweise das Registrieren der Windowklasse) direkt an das ursprüngliche Betriebssystem gegeben werden. Wenn allerdings ein API-Aufruf mit dem Windowmanagement in Beziehung steht (beispielsweise das Erzeugen von Fenstern, das Darstellen von Fenstern), fängt das Übersetzungsverfahren den API-Aufruf ab, bevor es den Aufruf zu dem ursprünglichen Betriebssystem weitergibt (Schritte 610 und 620). In dem Fall, in dem API CreateWindow aufruft, ruft die bevorzugte Ausführungsform entweder XGet-Geometry oder XGetWindow-Attribute auf, um die Dimensionen der computerfernen Server-Anzeigeeinrichtung zu bestimmen. Wenn die Dimensionen der computerfernen Serveranzeigeeinrichtung bekannt sind, ruft das Übersetzungsverfahren die Xlib-Funktion XCreateWindow, um ein computerfern dargestelltes Window in einer X-Umgebung zu erzeugen. In ähnlicher Weise stellt das Übersetzungsverfahren, wenn das Anwendungsprogramm einen ShowWindow-API-Aufruf ausgibt, ein entsprechendes X-Window (Schritt 620) dar, bevor es den ShowWindow-API-Aufruf an das ursprüngliche Betriebssystem leitet. Somit stellen die Beispiele in Fig. 6, Schritte 610 und 620, die abgefangene Meldung auf höherer Ebene, die im Zusammenhang mit Schritt 315 in

Fig. 3 diskutiert wurde, dar. Analog stellen die Schritte 630 und 640 in Fig. 6 die abgefangene Meldung auf niedriger Ebene, die in bezug auf Schritt 345 in Fig. 3 diskutiert wurde, dar. Insbesondere fängt das Übersetzungsverfahren die Aufrufe des Anwendungsprogramms an den Bildschirmanzeigeeinrichtungstreiber ab (Schritt 630). Wie oben erläutert, behandelt der Bildschirmtreiber keine Konstruktionen auf hoher Ebene, wie beispielsweise Fenster. Als ein Ergebnis muß das Übersetzungsverfahren zuerst das entsprechende X-Window bestimmen, für das der Aufruf zum Bildschirmtreiber ausgeführt wurde. Diese Bestimmung kann durch eine Anzahl von Wegen durchgeführt werden. Beispielsweise kann diese Bestimmung dadurch durchgeführt werden, daß die Datenstruktur, die die interne Windowzustandsinformation für alle X-Windows, die derzeit existieren, enthalten, aufrechterhalten und bewerten. In diesem Szenario bestimmt das Übersetzungsverfahren die Bildschirmkoordinaten, auf die durch den Bildschirmtreiber Bezug genommen wird. Das Übersetzungsverfahren kann dann auf diese Koordinaten verweisen, um eine Handhabe für die Windowstruktur, die diese Koordinaten besitzt, zu erhalten. Auf diese Handhabe kann dann seinerseits querverwiesen werden, um das entsprechende X-Window zu bestimmen. Nachdem das erwünschte X-Window bestimmt ist, übersetzt das Übersetzungsverfahren die Bildverarbeitungsanfragen in X-Aufrufe und sendet diese Aufrufe zum computerfernen X-Server. Auf diese Weise wird die computerferne Anzeigeeinrichtung schnell und effizient aktualisiert. Schließlich gibt das Übersetzungsverfahren optional den Bildschirmanzeigetreiberanruf an den lokalen Bildschirmtreiber, falls erwünscht. Dieser Schritt ist nur notwendig, wenn eine lokale Bildaufbereitung erwünscht ist.

Wie in dieser bevorzugten Ausführungsform erklärt, ermöglicht die vorliegende Erfindung, daß mit Anwendungsprogrammen, die unter dem "WINDOWS/NT"-Betriebssystem ausgeführt werden, computerfern über das X-Client/Server-Windowssystem wechselgewirkt wird. Eine besonders vorteilhafte Anwendung der bevorzugten Ausführungsform betrifft das Gebiet der Medizintechnik. Insbesondere liegt eine bevorzugte Ausführungsform der vorliegenden Erfindung in der Multi-Disclosure Review Station, hergestellt durch SpaceLabs, Incorporated, die einen medizinischen Monitor als X-Server verwendet. Dieser spezialisierte X-Server bereitet die kritischen Daten des Patienten (beispielsweise eine Herzwellenform) auf einem begleitenden X-Terminal zum Bild auf. Mit der vorliegenden Erfindung kann der Arzt nicht nur die Herzwellenform eines Patienten auf diesem Terminal ansehen, sondern der Arzt kann auch auf Anwendungsprogramme, die für das "WINDOWS/NT"-Betriebssystem geschrieben sind, zugreifen. Als Ergebnis wird die Arbeit des Arztes stark vereinfacht. Beispielsweise kann eine Anwendung, wie z. B. "MICROSOFT EXCEL" verwendet werden, um eine Datenbank der Krankengeschichte des Patienten zu erstellen und zu aktualisieren. Somit erlaubt die vorliegende Erfindung, daß der Arzt die aktualisierte Krankengeschichte des Patienten abrufen, während er simultan die Herzwellenform des Patienten ansieht. Da die vorliegende Erfindung gleichzeitigen Zugriff zu momentanen und vorangegangenen medizinischen Bedingungen schafft, kann der Arzt schnell und leicht alle Information erhalten, die notwendig ist, um den Patienten über ein einzelnes Terminal zu behandeln. Auf diese Weise kann der Arzt effizienter arbeiten.

Die oben gelieferte, detaillierte Diskussion stellt bevorzugte Ausführungsformen der vorliegenden Erfindung dar. Diese Diskussion ermöglicht es dem Fachmann, verschiedene Modifikationen dieser Ausführungsform durchzuführen, die nicht von dem Umfang der Erfindung abweichen. Beispielsweise wird, obwohl eine bevorzugte Ausführungsform unter Bezugnahme auf das "WINDOWS/NT"-Betriebssystem beschrieben wurde, der Fachmann erkennen, daß das hierin offenbarte erfinderische Konzept nicht auf irgendein spezielles Betriebssystem beschränkt ist. In ähnlicher Weise wird der Fachmann, obwohl die bevorzugte Ausführungsform unter Bezugnahme auf das X-Protokoll beschrieben wurde, erkennen, daß die hierin offenbarten erfinderischen Konzepte nicht auf das Protokoll irgendeiner bestimmten, computerfernen grafischen Benutzeroberfläche beschränkt sind. Demgemäß berücksichtigt die vorliegende Erfindung alle derartigen Modifikationen, die sich auf die angefügten Ansprüche und Äquivalente derselben lesen.

Patentansprüche

1. Ein Verfahren, um eine Anwendung als X-Client auszuführen, so daß ermöglicht wird, die Anwendung mit einem X-Windowmanager darzustellen, wobei das Verfahren die Schritte aufweist:
Starten einer "WINDOWS/NT"-Sitzung;
Aufrufen einer Anwendung aus der "WINDOWS/NT"-Sitzung;
Überwachen der Ausgabenachrichten, die von der Anwendung gesendet werden;
Bestimmen, wann eine der überwachten Ausgabenachrichten ein Befehl einer grafischen Benutzeroberfläche ist; und,
wenn die überwachte Ausgabenachricht ein Befehl einer grafischen Benutzeroberfläche ist, Suspendieren eines "WINDOWS/NT"-Gerätetreibers vom Handeln auf den Befehl hin, Handelnlassen eines X-Windowmanager-Programms auf den Befehl hin, und Informieren des "WINDOWS/NT"-Gerätetreibers über das Ergebnis des Befehls, auf den durch das X-Windowmanager-Programm hin gehandelt worden ist.
2. Das Verfahren nach Anspruch 1, weiter umfassend den Schritt:
Konvertieren des Befehls von einem "WINDOWS/NT"-Format in ein Format, das durch einen X-Server interpretierbar ist, so daß dem X-Server ermöglicht wird, auf den Befehl hin zu handeln.
3. Ein Verfahren zum Bereitstellen eines computerfernen Zugriffs auf ein Anwendungsprogramm, das unter einem ursprünglichen Betriebssystem ausgeführt wird, in welchem das ursprüngliche Betriebssystem eine Anwendungsschnittstelle zur Verfügung stellt, um Befehle vom Anwendungsprogramm zu empfangen, und in welchem das Betriebssystem eine ursprüngliche grafische Benutzeroberfläche zur Verfügung stellt, wobei das Verfahren die Schritte aufweist:
Überwachen einer Nachricht, die vom Anwendungsprogramm an die Anwendungsschnittstelle ausgegeben wird;
Abfangen der überwachten Nachricht, wenn die Nachricht eine Operation ist, die die ursprüngliche grafische Benutzeroberfläche beeinflusst; und
Umleiten der abgefangenen Nachricht auf einen nicht-ursprünglichen grafischen Benutzeroberfläche.

chenmanager, so daß dem nicht-ursprünglichen grafischen Benutzeroberflächenmanager ermöglicht wird, die Operation der überwachten Nachricht durchzuführen.

4. Das Verfahren nach Anspruch 3, weiter umfassend den Schritt:

Informieren des ursprünglichen Betriebssystems über die Ergebnisse der abgefangenen Nachricht.

5. Ein Verfahren zum Bereitstellen einer computerfernen Anzeige und Interaktion mit einem Anwendungsprogramm, das auf einer lokalen Maschine ausgeführt wird, wobei die lokale Maschine die Anwendung in einer Umgebung eines ursprünglichen Betriebssystems ausführt, in welchem das ursprüngliche Betriebssystem eine lokale Anwendungsschnittstelle, um Befehle vom Anwendungsprogramm zu empfangen, und eine lokale grafische Benutzeroberfläche zur Verfügung stellt, wobei das Verfahren die Schritte aufweist:

Erkennen eines Befehls, der vom Anwendungsprogramm an die Anwendungsschnittstelle ausgegeben wird; und

Konvertieren der erkannten Nachricht von einem Format für die lokale grafische Benutzeroberfläche in ein Format für eine computerferne grafische Benutzeroberfläche, so daß die erkannte Nachricht durch eine computerferne grafische Benutzeroberfläche interpretiert werden kann.

6. Ein Verfahren zum Bereitstellen einer computerfernen Benutzerinteraktion mit einem Anwendungsprogramm, das in seiner ursprünglichen Umgebung auf einer Hauptmaschine ausgeführt wird, wobei die ursprüngliche Umgebung eine lokale grafische Benutzeroberfläche zur Verfügung stellt, die eine lokale Benutzerinteraktion mit dem Anwendungsprogramm erlaubt, wobei das Verfahren die Schritte aufweist:

Bestimmen, wann das Anwendungsprogramm einen Befehl an die lokale grafische Benutzeroberfläche gesendet hat;

Suspendieren der Fähigkeit der ursprünglichen Umgebung, den bestimmten Befehl zu verarbeiten; und

Konvertieren des bestimmten Befehls in ein Protokoll, das von einer computerfernen grafischen Oberfläche erkennbar ist.

7. Eine Vorrichtung, die es erlaubt, ein Anwendungsprogramm wie einen computerfernen X-Clienten zu behandeln, während das Anwendungsprogramm in einer Nicht-Unix-Umgebung abläuft, wobei die Vorrichtung aufweist:

eine Überwachungseinrichtung, die bestimmt, wann das Anwendungsprogramm einen Befehl an eine lokale grafische Benutzeroberfläche gesendet hat, wobei die lokale grafische Benutzeroberfläche durch die Nicht-Unix-Umgebung zur Verfügung gestellt wird;

eine Deaktivierungseinrichtung, die die Fähigkeit der Nicht-Unix-Umgebung, den bestimmten Befehl zu verarbeiten, deaktiviert; und

eine Modifiziereinrichtung, die den bestimmten Befehl in ein Protokoll, das von einer computerfernen grafischen Oberfläche erkennbar ist, konvertiert.

8. Ein Verfahren, das es erlaubt, eine Windows- oder "WINDOWS/NT"-Anwendung, zusätzlich zu ihrer Ausführung als X-Client, in ihrer ursprünglichen Umgebung auszuführen, so daß ermöglicht wird, die Anwendung auf einem computerfernen

X-Terminal darzustellen, wobei das Verfahren die Schritte aufweist:

computerfernes Starten einer Windows- oder "WINDOWS/NT"-Anwendung;

Überwachen von System-Aufrufen und Anzeigeneinrichtungstreiber-Aufrufen, die von der Windows- oder der "WINDOWS/NT"-Anwendung ausgeführt werden;

Bestimmen, wann einer der überwachten System-Aufrufe ein mit einer grafischen Benutzeroberfläche zusammenhängender Befehl ist;

Bestimmen, wann eine Bildschirm-Bildaufbereitung für ein Fenster, das der Windows- oder der "WINDOWS/NT"-Anwendung zugeordnet ist, bestimmt ist; und,

jedesmal wenn überwachte System-Aufrufe Befehle einer grafischen Benutzeroberfläche sind und jedesmal wenn eine Anzeigeneinrichtungstreiber-Bildaufbereitung auf ein Fenster, das dem Windows- oder dem "WINDOWS/NT"-Anwendungsprogramm zugeordnet ist, abzielt, Ausführen geeigneter X-System-Aufrufe, um die beabsichtigte Wirkung auf dem computerfernen X-Terminal zu erzielen, und dann Zulassen, daß ein Betriebssystem der ursprünglichen Umgebung den Aufruf verarbeitet.

9. Ein Verfahren zur Laufzeitübersetzung von einem ersten grafischen Betriebssystem in ein zweites grafisches Betriebssystem, in welchem das erste grafische Betriebssystem keine computerferne Darstellung von Anwendungsprogrammen vorsieht und in welchem das zweite grafische Betriebssystem eine computerferne Darstellung von Anwendungsprogrammen vorsieht, wobei das Verfahren die Schritte aufweist:

Überwachen von System-Aufrufen und Anzeigeneinrichtungstreiber-Aufrufen, die von einem Anwendungsprogramm, das unter dem ersten grafischen Betriebssystem ausgeführt wird, durchgeführt werden;

Bestimmen, wann einer der überwachten System-Aufrufe ein mit einer grafischen Benutzeroberfläche zusammenhängender Befehl ist;

Bestimmen, wann eine Bildschirm-Bildaufbereitung auf ein Fenster abzielt, das dem Anwendungsprogramm, das unter dem ersten grafischen Betriebssystem ausgeführt wird, zugeordnet ist, und,

jedesmal wenn überwachte System-Aufrufe Befehle einer grafischen Benutzeroberfläche sind und jedesmal wenn eine Anzeigeneinrichtungstreiber-Bildaufbereitung auf ein Fenster abzielt, das dem Anwendungsprogramm, das unter dem ersten grafischen Betriebssystem ausgeführt wird, zugeordnet ist, Ausführen geeigneter Befehle in dem zweiten grafischen Betriebssystem, um die beabsichtigte Wirkung auf einer computerfernen Anzeigeneinrichtung, die vom zweiten grafischen Betriebssystem gesteuert wird, zu erzielen, und dann Zulassen, daß das erste grafische Betriebssystem den Aufruf verarbeitet.

10. Ein Verfahren zum Übersetzen von Befehlen eines Anwendungsprogramms aus einem ersten grafischen Betriebssystem in ein zweites grafisches Betriebssystem, wobei das Anwendungsprogramm in der ersten grafischen Umgebung ausgeführt wird, in welchem das erste grafische Betriebssystem keine computerferne Darstellung des Anwendungsprogramms vorsieht und in welchem das

zweite grafische Betriebssystem eine computerferne Darstellung des Anwendungsprogramms vorsieht, wobei das Verfahren die Schritte aufweist: Überwachen von Gerätetreiber-Aufrufen, die durch das Anwendungsprogramm ausgeführt werden;
Bestimmen, wann einer der überwachten Gerätetreiber-Aufrufe auf ein Fenster, das dem Anwendungsprogramm zugeordnet ist, abzielt, und, jedesmal wenn einer der überwachten Gerätetreiber-Aufrufe auf ein Fenster, das dem Anwendungsprogramm zugeordnet ist, abzielt, Ausgeben geeigneter Befehle in dem zweiten grafischen Betriebssystem, um die beabsichtigte Wirkung auf einer computerfernen Anzeigeeinrichtung, die vom zweiten grafischen Betriebssystem gesteuert wird, zu erzielen.

Hierzu 6 Seite(n) Zeichnungen

20

25

30

35

40

45

50

55

60

65

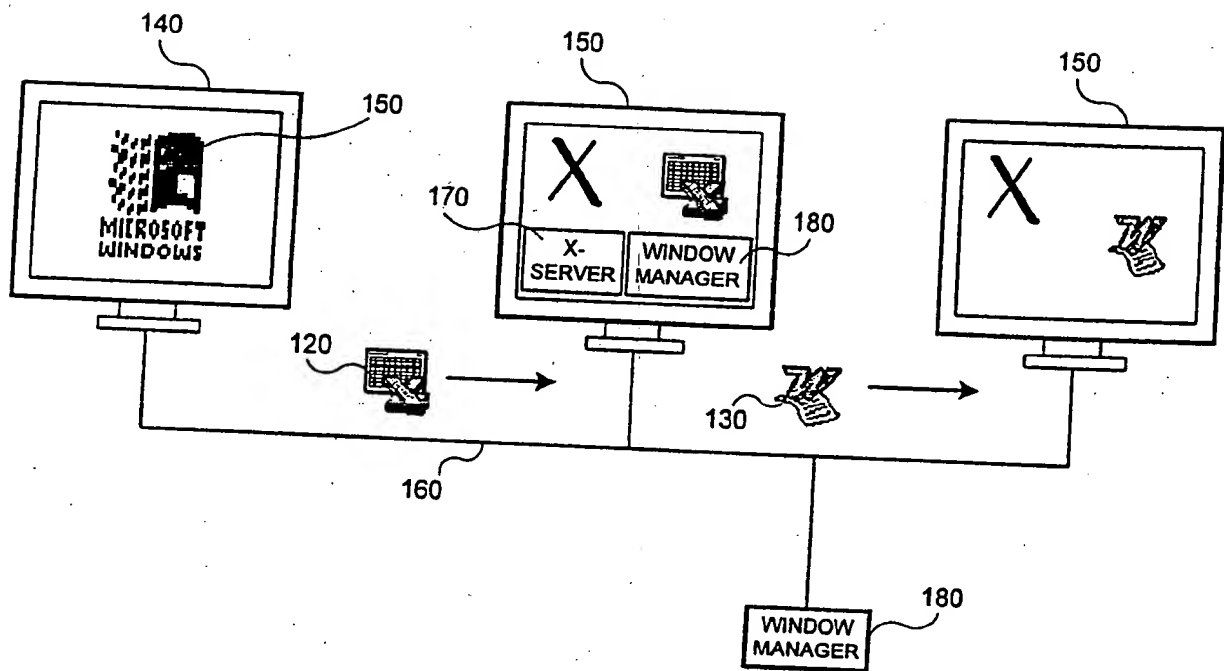


FIG. 1

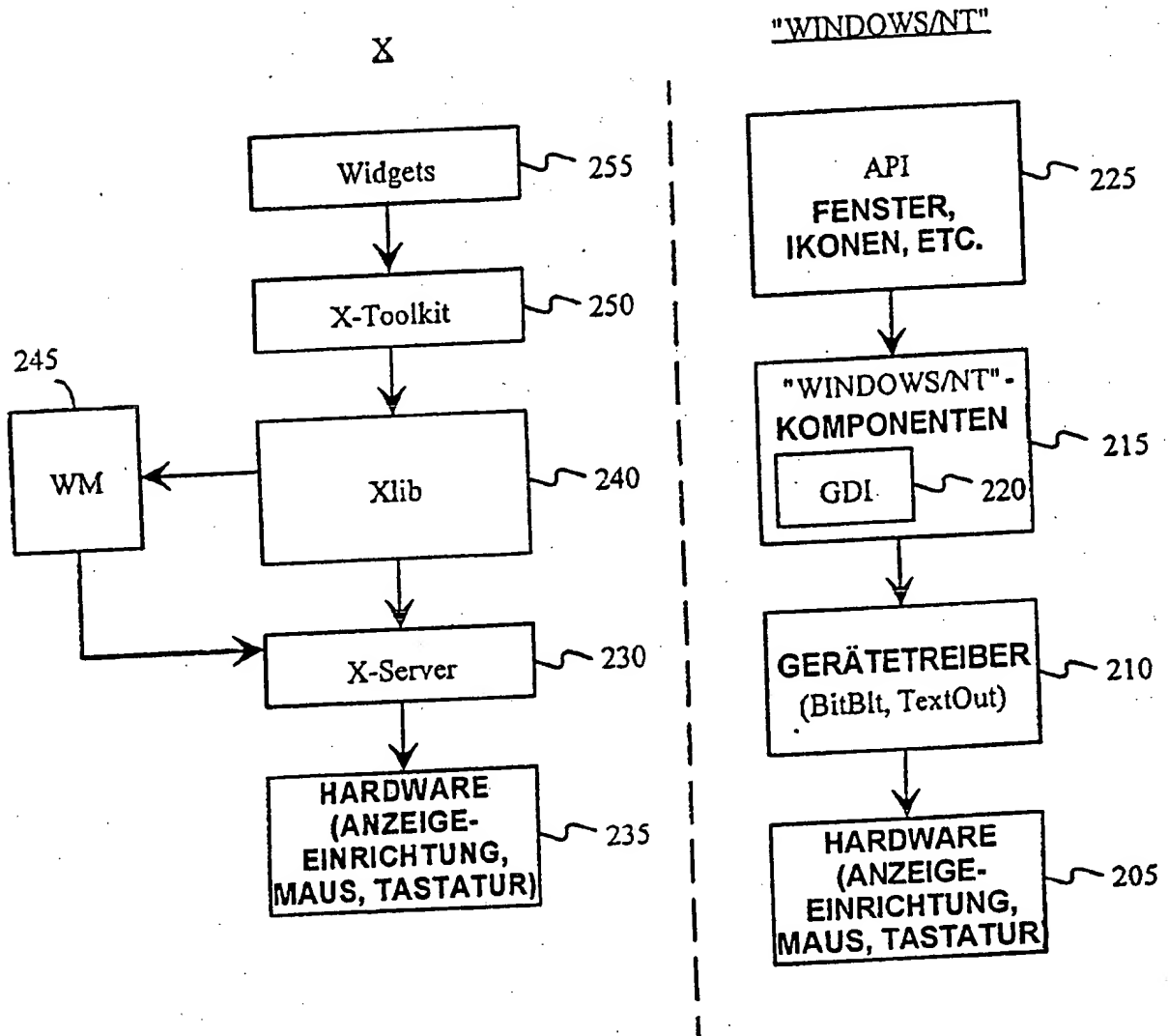


FIG. 2
STAND DER TECHNIK

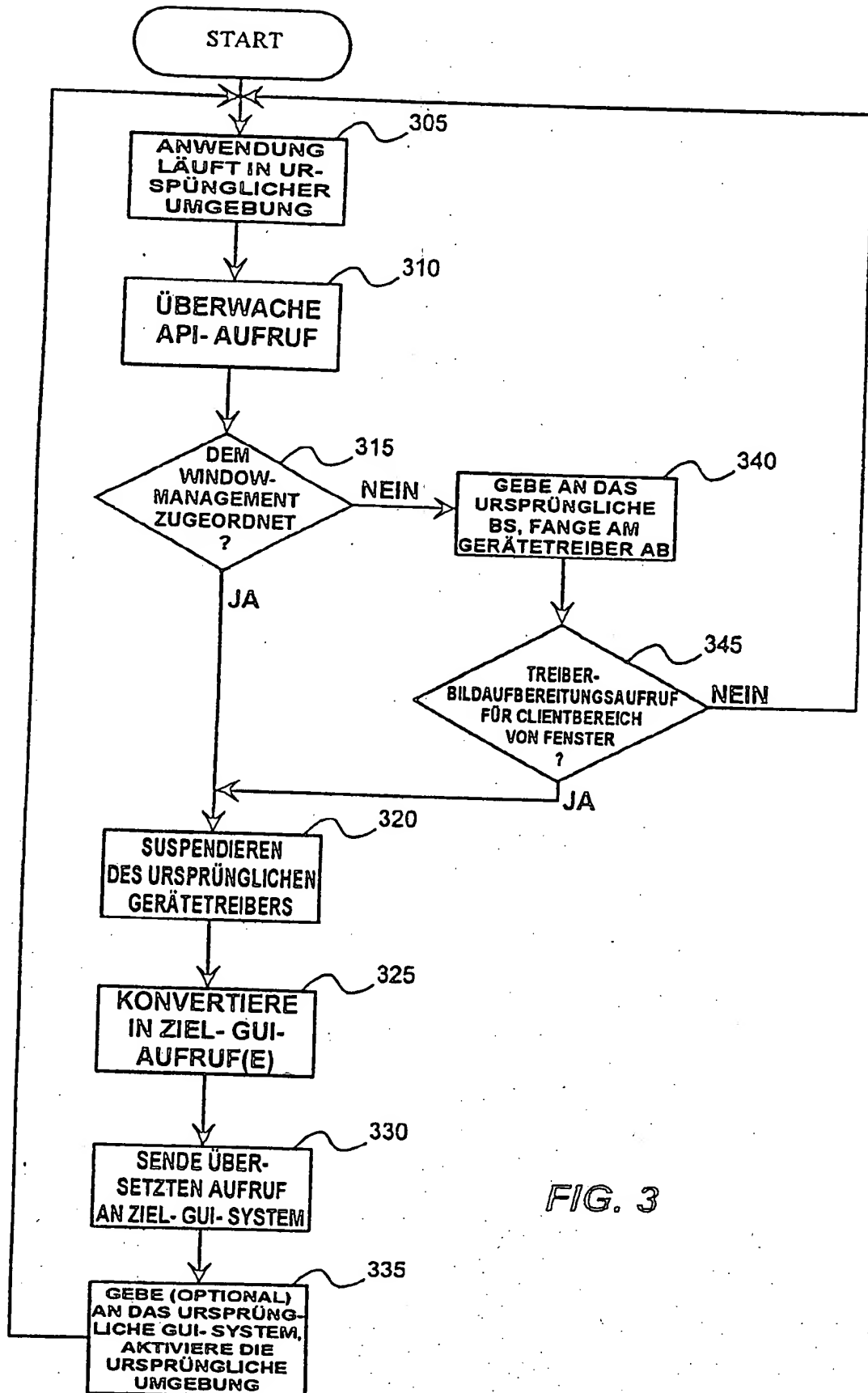


FIG. 3

PROGRAMM:

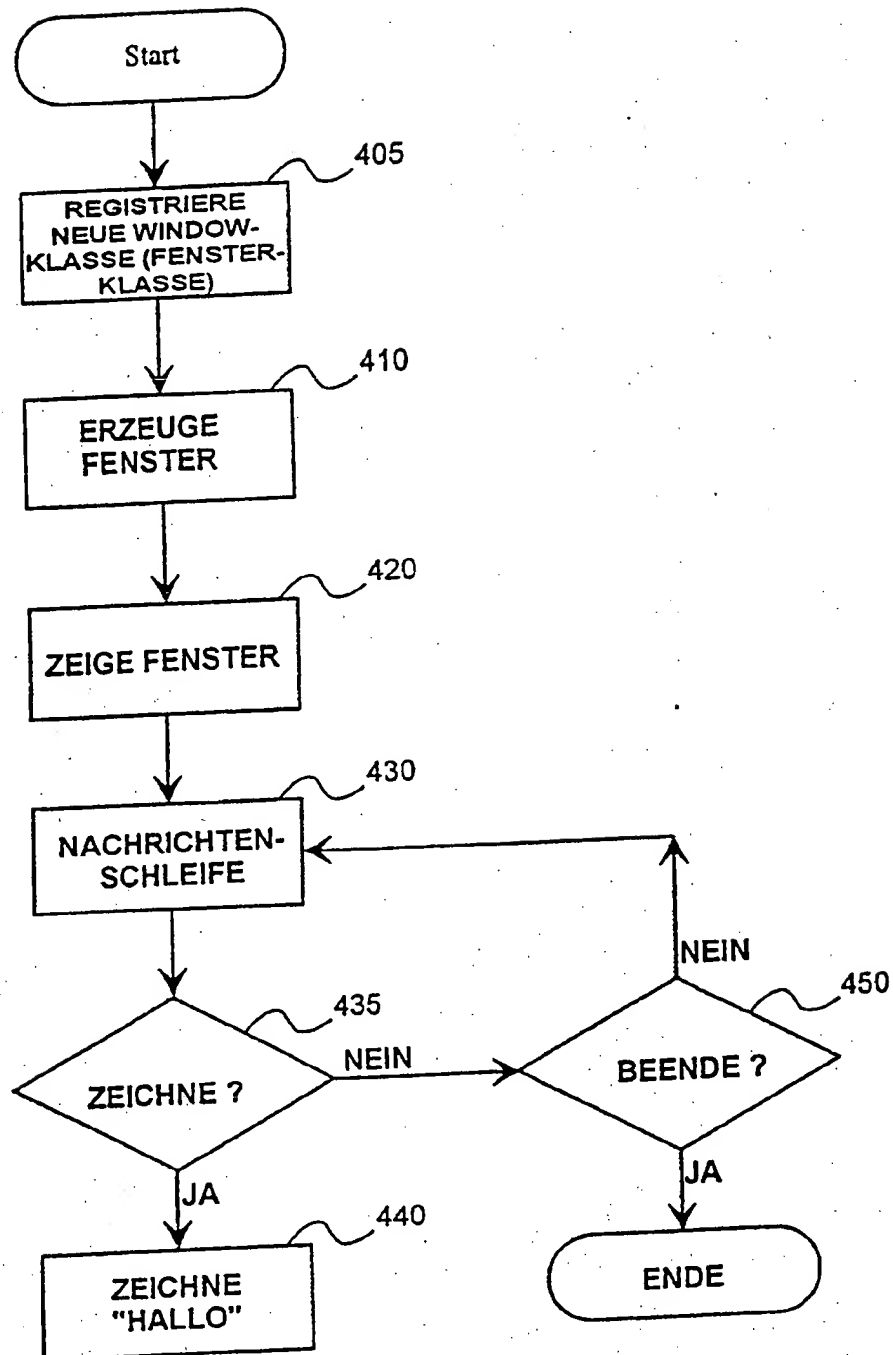


FIG. 4
STAND DER TECHNIK

PROGRAMM:

URSPRÜNGLICHES BS:

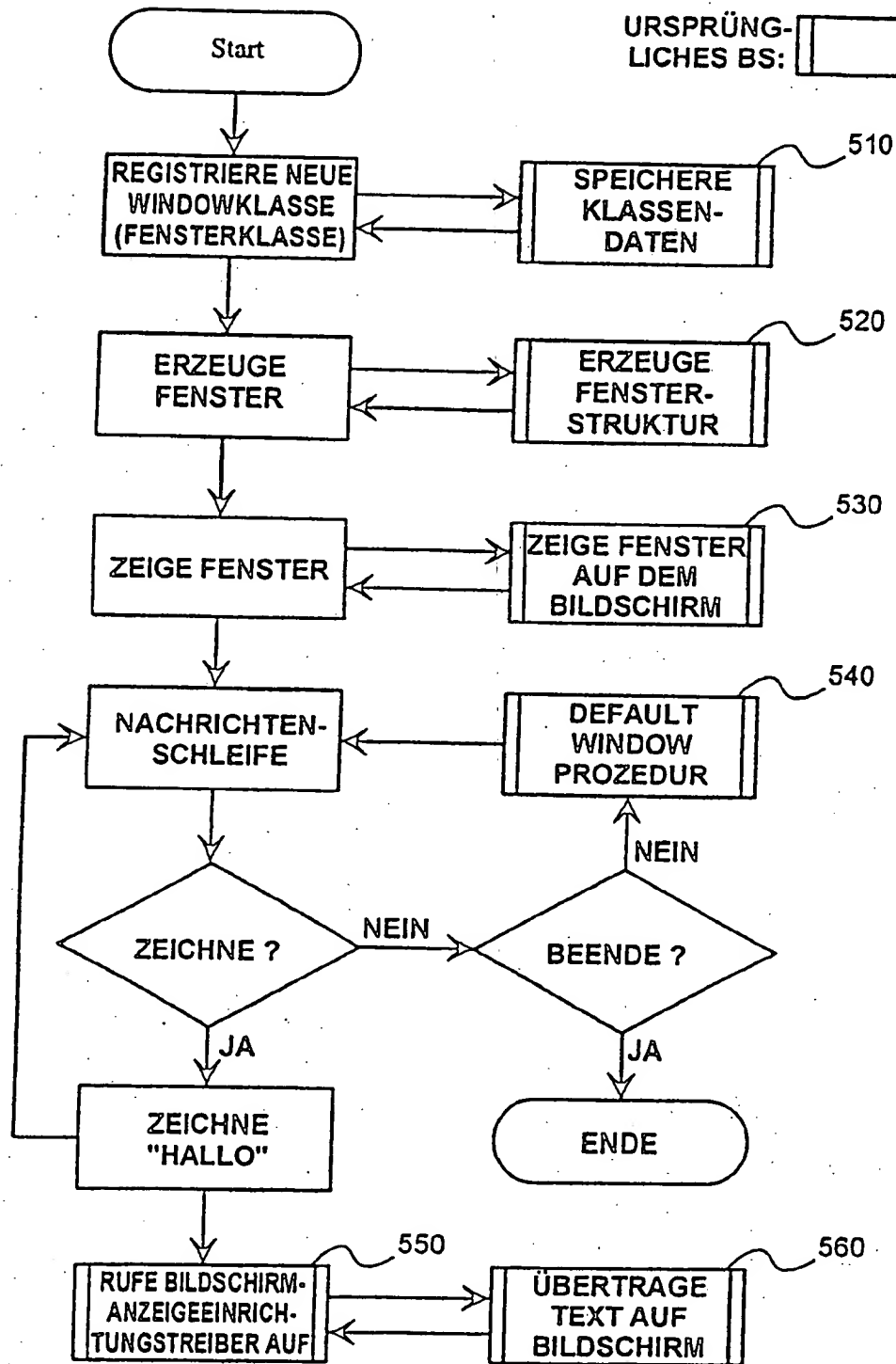


FIG. 5
 STAND DER TECHNIK

PROGRAMM:

URSPRÜNGLICHES BS:

ÜBERSETZUNGSSYSTEM:

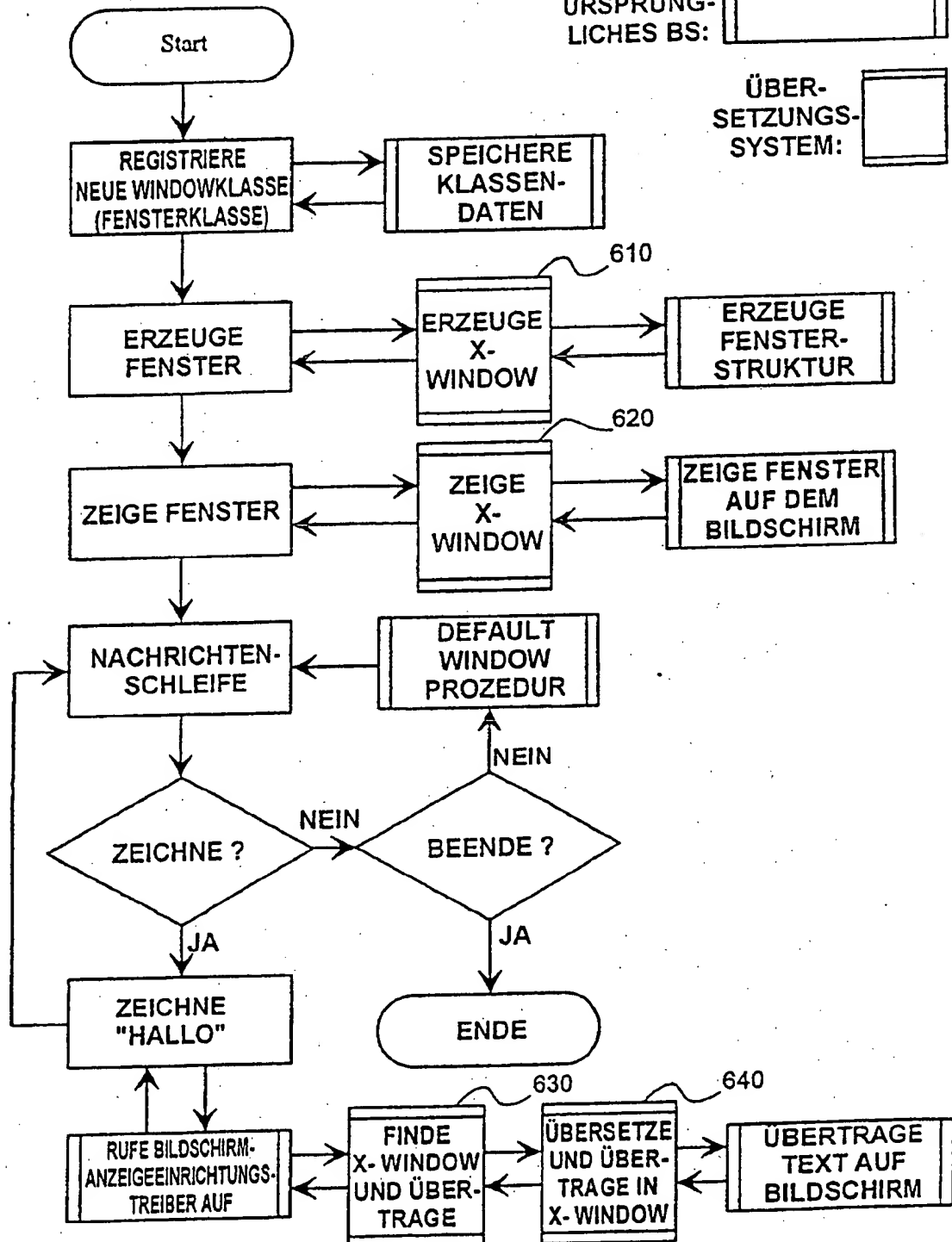


FIG. 6